

How to Handle Alerts and Pop-ups in Selenium?

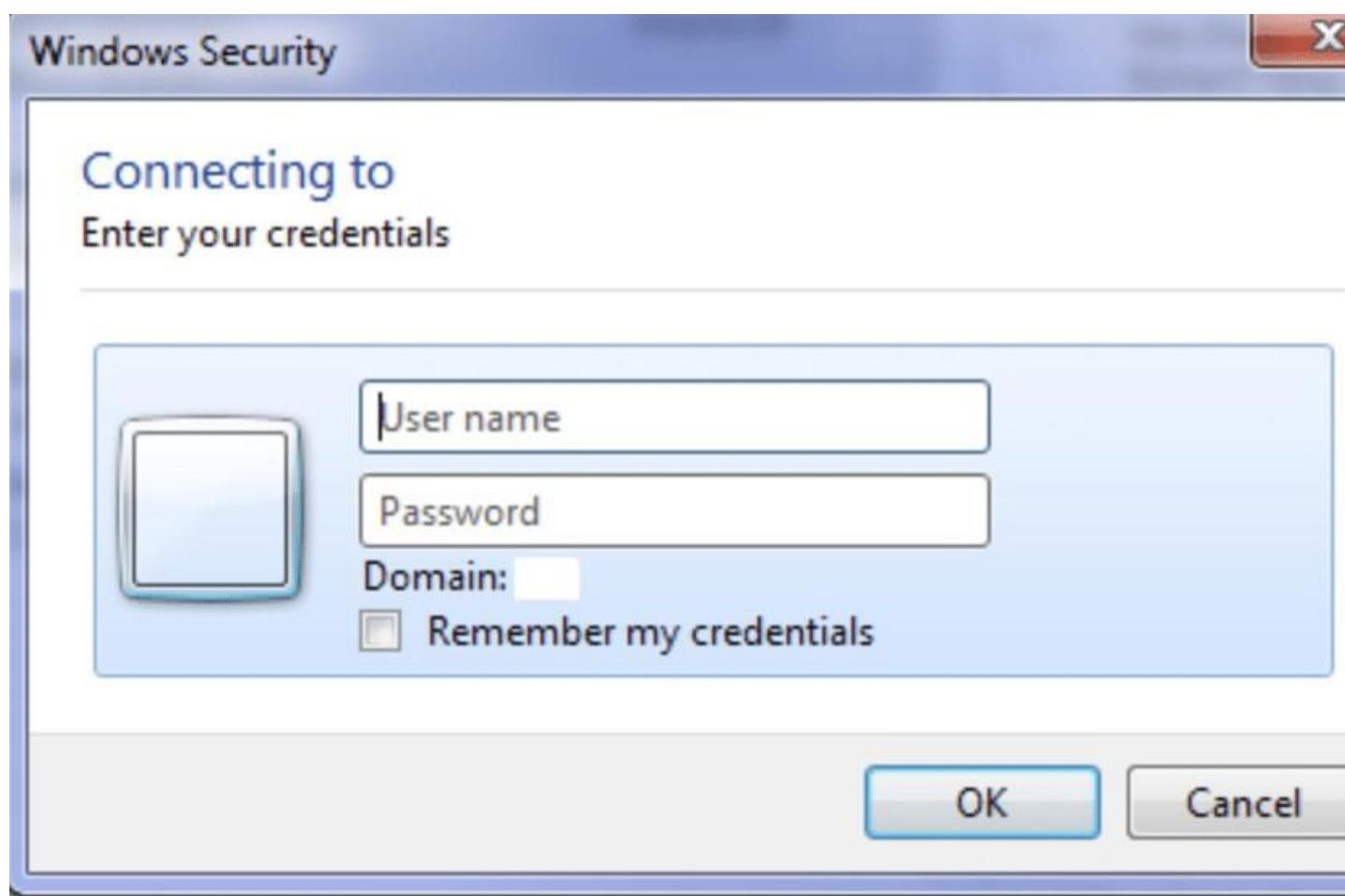
Imagine that you are filling an application form and missing some details by mistake. How do you come to know this until there is an alert or a pop-up window to tell you? Let's overview the following article to understand what are alerts in selenium and how to handle and manage them. An Alert in selenium is any message or notification box that informs the user about some information or asks for permission to perform particular tasks. It is also used for warning purposes.



There are three types of Alerts in Selenium–

1. Simple Alert

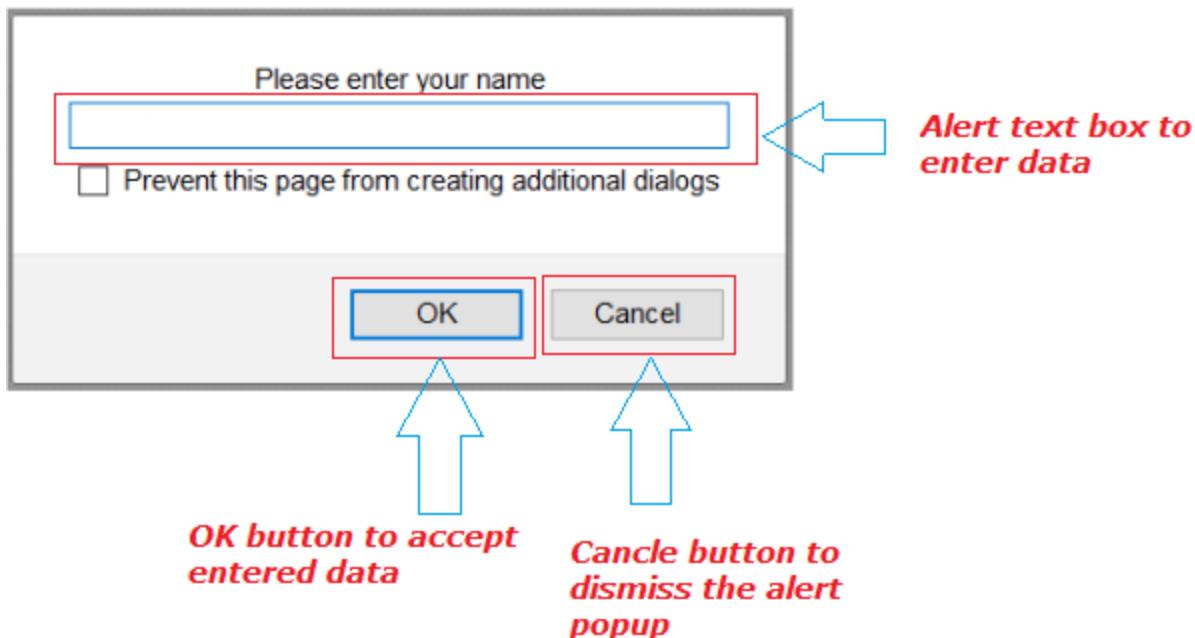
The simple alert class in Selenium presents some information or notify a warning message on the screen.



2. Prompt Alert

This Prompt Alert asks for some input from the user about the required information to complete the task. Selenium web drivers can input the text with the use of sendkeys("input....").

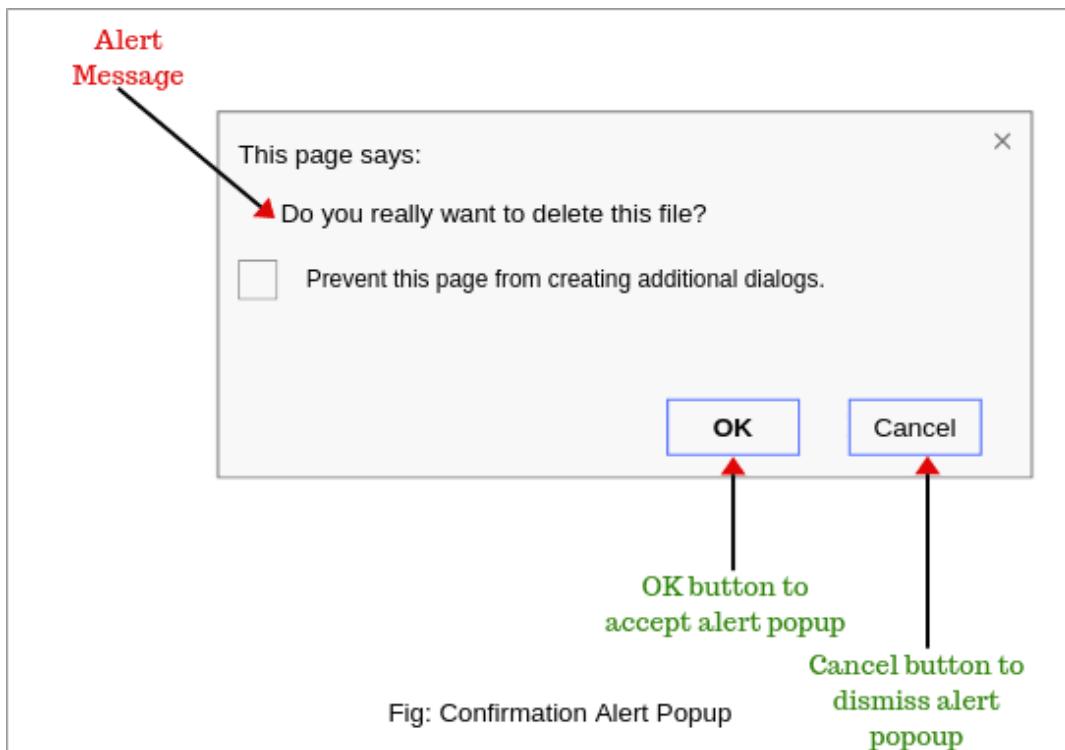
Prompt dialogue box



3. Confirmation Alert

A confirmation alert requires permission to do some operation or to perform a particular task. This alert is used for the confirmation of some operations or tasks.

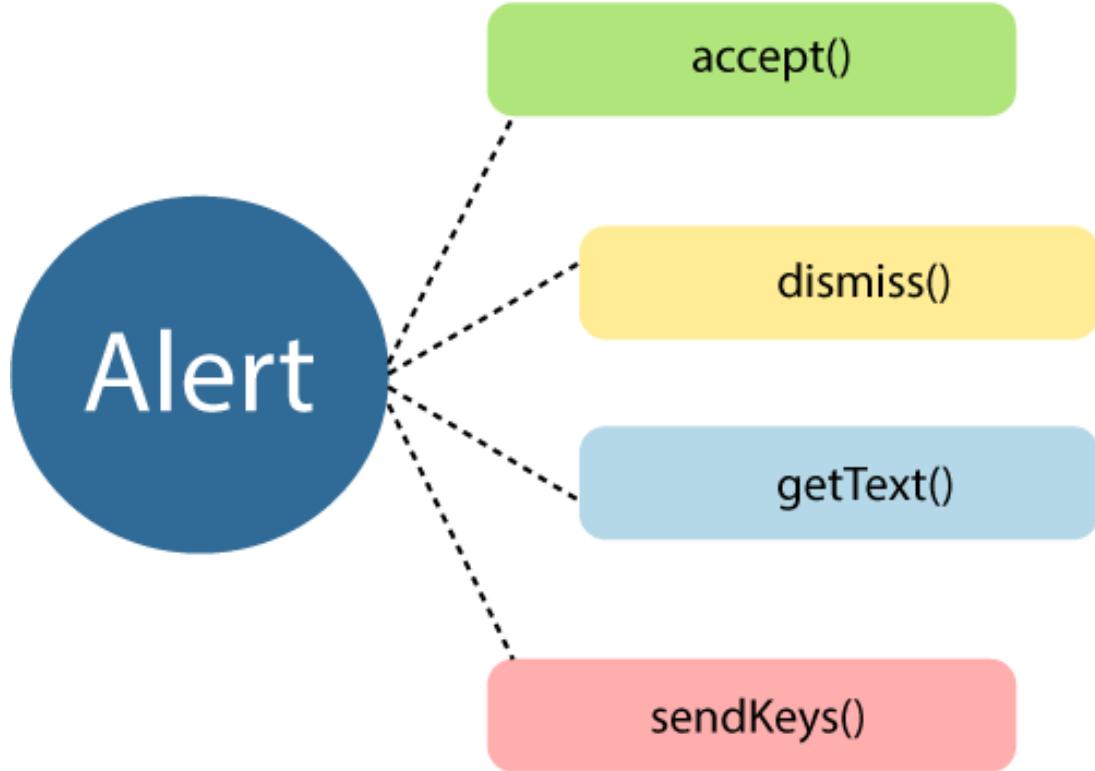
For Example: Do you want to proceed with a particular task? Yes or No?



Alert Handling in Selenium

Handling alerts with one's hands is a tedious task. Selenium offers a good range of methods to handle and manage alerts to decrease human interference and reduce this task.

Methods of handling alerts in Selenium



Void dismiss()– This method is applied when the 'Cancel' button is clicked in the alert box.

Syntax– `driver.switchTo().alert().dismiss();`

Void accept()– This method is used to click on the 'OK' button of the alert. Syntax–

`driver.switchTo().alert().accept();`

String getText(): This process is used to capture the alert messages Syntax–

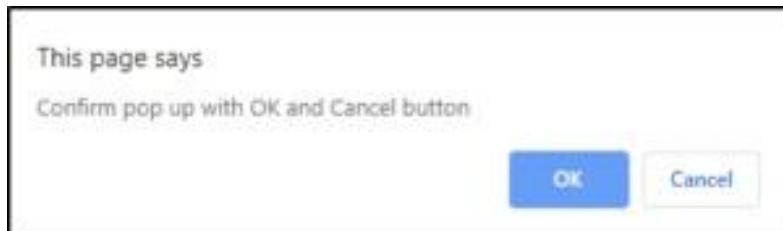
`driver.switchTo().alert().getText();`

Void sendKeys(String stringToSend): This process is used to send some data to the alert box

Syntax– `driver.switchTo().alert().sendKeys("Text");`

Popups in Selenium

Popup is a window that presents or pops up on the screen because of some activity. In computerization, when we have multiple windows in a web application, any action may need to switch control among various windows from one to another to complete the operation or task. After the process is completed, it has to return to the main window, i.e., the parent window in Selenium.



How to handle popups in Selenium

In the selenium web driver, there are multiple methods to handle popups:

1. `Driver.getWindowHandles();`

To handle all the opened windows by Selenium web driver, you can use

`Driver.getWindowHandles()` to switch from one window to another in a web application.

2. `Driver.getWindowHandle();`

When the webpage is loaded, or the site opens, you can handle the main window by using `the driver.getWindowHandle()`. This will handle the current window that is uniquely recognized within the driver instance.

Handling Popup Window using Selenium

In the Selenium web driver, the robot class handles and manages all the keyboard and mouse functions. In addition, it is used to close the pop-up window. You can get the window handle out of the pop-up window using the *WindowHandle()* function.

Handling Unexpected Alerts using Selenium WebDriver

When we are browsing a web application, unexpected alerts occur due to some error or other reasons. This alert is not displayed every time you open the site but they appear only at random intervals. If you have built any automation test case for some pages and have not handled this kind of alert, then your script will fail instantly if such an unexpected alert pop-up is presented.

Conclusion-

Hence Alerts or Pop-Ups are necessary for web applications. While filling any form, if any alert or popup is displayed on the screen, handle it very effectively using the methods mentioned above of alerts handling or popup handling process.